

1. 업무 분석 및 정의서

회원정보 관리

학교 구성원인 교수와 학생들은 사이트를 이용하고자 회원가입을 한다.

회원가입을 하면서 입력한 정보인 식별번호, 아이디, 비밀번호, 가입일자가 회원정보로 저장된다.

이때 식별번호는 학교에서 교수와 학생들을 식별하고자 부여한 번호이다.

이 번호는 쉽게 말하면 학생 입장에서는 학번이 되고, 교수 입장에서는 사번과 같은 존재이다.

이때 교수의 사번은 알파벳 p로 시작하므로 학생과 구분할 수 있다.

즉 이 번호는 중복되어서는 안된다.

또한 회원가입을 할 때 구성원들의 아이디는 모두 다르다.

학생, 교수 정보 관리

회원정보를 바탕으로 학생정보와 교수 정보를 따로 보관한다.

회원가입할 때 입력한 식별번호를 바탕으로 데이터를 추가할 수 있다.

이는 즉 회원정보가 교수와 학생정보를 포함한다고 볼 수 있다.

학생 입장에서 식별번호는 학번이다.

각 학생들의 학번은 모두 다르다.

학생 정보로 학번, 이름, 학과를 저장한다.

교수 입장에서 식별번호는 사번이다.

각 교수들의 사번은 모두 다르다.

교수 정보로 사번, 이름을 저장한다.

과목 정보 관리

과목은 과목코드, 과목명, 교수명, 수강료 정보를 갖고 있다.

과목코드는 중복되지 않는다.

수업

한 교수는 여러 과목을 수업할 수 있지만, 한 과목을 여러 교수가 수업할 수는 없다.

주차별 과제, 강의 정보

각 과목에는 과제와 강의를 존재한다.

과목에 강의 혹은 과제가 존재하지 않아도 된다. 하지만 강의, 과제는 과목에 의해서만 생성된다.

하나의 과목에는 여러 개의 과제, 강의를 존재할 수 있다.

과제는 과목코드와 과제코드, 시작일, 마감일, 과제내용 정보를 담고 있다.

강의는 과목코드와 주차, 차시, 강의내용 정보를 담고 있다.

강의는 한 차시에 한 개만 올라온다.

수강신청

학생들은 과목을 수강하기 전에 수강신청을 해야한다.

수강신청은 각 과목의 수강료를 지불하는 방식으로 이루어진다.

수강료를 지불할 경우 그 과목은 수강신청 된 것이다.

모든 학생은 수강료를 지불할 충분한 돈을 지니고 있다. (즉 금전적 여유가 된다.)

이때 학생은 모두 재학생으로 수강신청을 반드시 해야한다.

한 학생은 여러 과목을 수강신청 할 수 있다.

하나의 과목은 여러 학생들에 의해 수강신청 될 수 있다.

강의 수강 기록

학생은 자신이 수강신청한 과목의 강의만 수강할 수 있다.

강의를 수강할 경우 수강기록이 남는데 과목코드, 주차, 차시, 학번, 수강여부가 기록된다.

이때 수강여부는 (수강, 지각, 결석, 미결)중 하나로 기록된다.

과제 수행 기록

학생은 자신이 수강신청한 과목의 과제만 수행할 수 있다.

한 학생은 여러 과제를 하고, 한 과제 또한 여러 명의 학생에 의해 수행됩니다.

과목의 과제를 할 경우 과제기록이 남는데 과목코드, 과제코드, 학번, 수행여부가 기록된다.

이때 수행여부는 (제출, 지각제출, 미제출)중 하나로 기록된다.

강의자료실에 글쓰기

교수는 강의자료실에 게시글을 작성할 수 있다.

게시글은 반드시 교수에 의해서만 작성된다.

강의자료실은 글번호, 글제목, 글내용, 작성일자, 작성자 정보를 지닌다.

2. 개념적 설계

2-1. 개체와 속성 추출

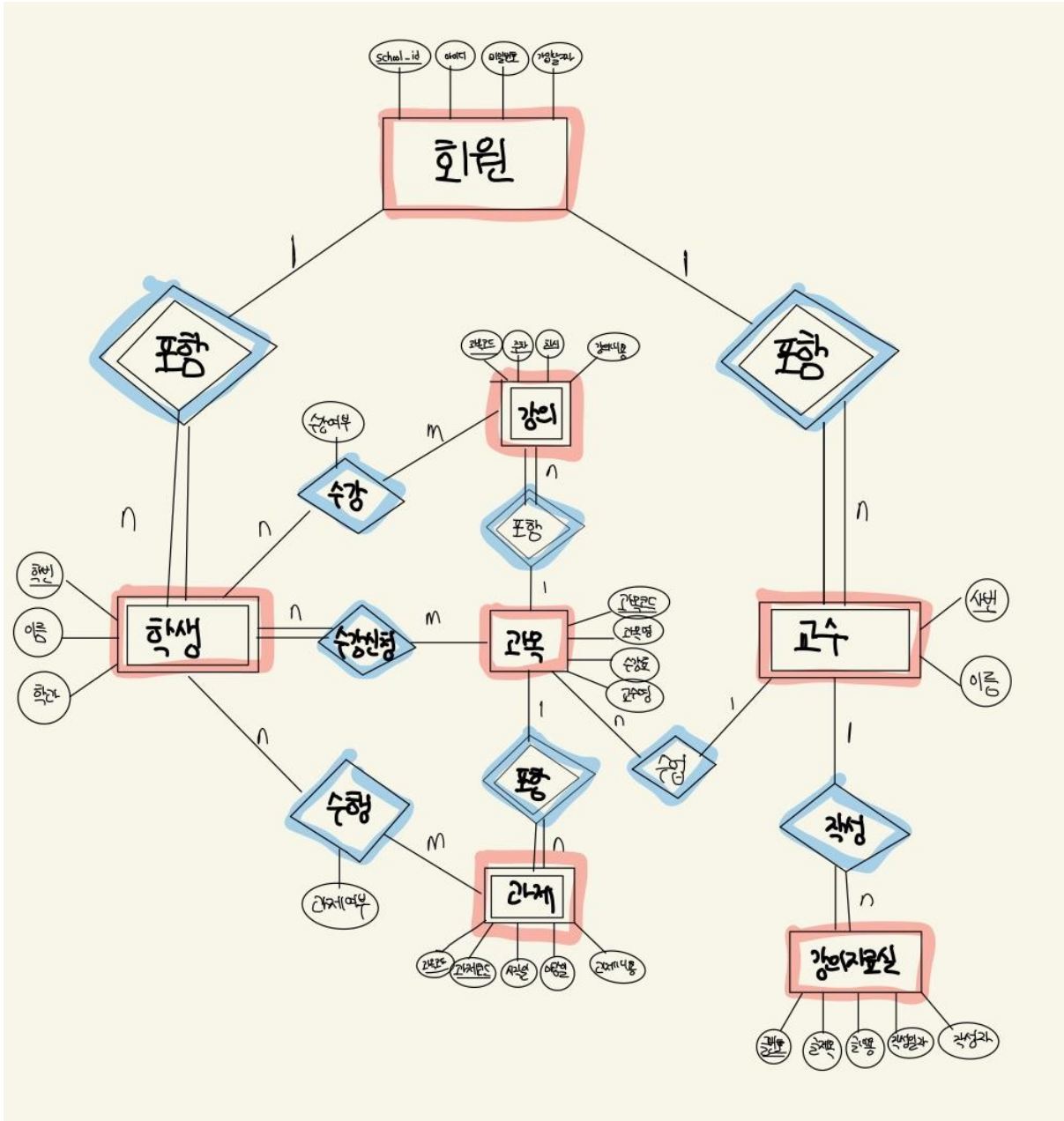
개체	속성
회원	<u>School_id</u> , 아이디, 비밀번호, 가입날짜
학생	<u>학번</u> , 이름, 학과
교수	<u>사번</u> , 이름
과목	<u>과목코드</u> , 과목명, 수강료, 교수명
강의	<u>과목코드</u> , 주차, 차시, 강의내용
과제	<u>과목코드</u> , <u>과제코드</u> , 시작일, 마감일, 과제내용
강의자료실	<u>글번호</u> , 글제목, 글내용, 작성일자, 작성자

2-2. 관계 추출

관계	관계에 참여하는 개체	관계 유형	속성
포함	회원(강한 개체), 학생(약한 개체)	일대다 (식별관계)	
포함	회원(강한 개체), 교수(약한 개체)	일대다 (식별관계)	
수강신청	학생(필수), 과목(선택)	다대다	
수업	과목(선택), 교수(선택)	일대다	
포함	과목(강한 개체), 강의(약한 개체)	일대다 (식별관계)	
포함	과목(강한 개체), 과제(약한 개체)	일대다 (식별관계)	

수강	학생(선택), 강의(선택)	다대다	수강여부
수행	학생(선택), 과제(선택)	다대다	수행여부
작성	교수(선택), 강의자료실(필수)	일대다	

2-3. E-R 다이어그램



3. 논리적 모델

3-1. 릴레이션 스키마로 변환

규칙 1 : 모든 개체는 릴레이션으로 변환한다.

- 회원 릴레이션

School_id	아이디	비밀번호	가입날짜
-----------	-----	------	------

- 학생 릴레이션

학번	이름	학과
----	----	----

- 교수 릴레이션

사번	이름
----	----

- 과목 릴레이션

과목코드	과목명	교수명	수강료
------	-----	-----	-----

- 강의 릴레이션

과목코드	주차	차시	강의내용
------	----	----	------

- 과제 릴레이션

과목코드	과제코드	시작일	마감일	과제내용
------	------	-----	-----	------

- 강의자료실 릴레이션

글번호	글제목	글내용	작성일자	작성자
-----	-----	-----	------	-----

규칙 2 : 일대다 관계는 외래키로 표현한다.

(외래키는 기울임 꼴로 나타냈습니다.)

2-1) 일반적인 일대다 관계는 외래키로 표현.

- 수업 : 교수(1)와 과목(n) 관계

교수의 기본키(사번)를 과목의 릴레이션에 포함시켜 외래키로 지정, 속성명은 교수 사번으로 바꾼다.

→ 과목 릴레이션

기본키 : 과목코드, 외래키 : 사번

과목코드	과목명	교수명	수강료	교수사번
------	-----	-----	-----	------

- 작성 : 교수(1)와 강의자료실(n) 관계

교수의 기본키(사번)를 강의자료실의 릴레이션에 포함시켜 외래키로 지정

이때 강의자료실 릴레이션의 작성자 속성이 사번을 뜻하므로 속성명을 작성자로 바꾸고 외래키로 지정한다.

→ 강의자료실 릴레이션

기본키 : 글번호, 외래키 : 작성자

글번호	글제목	글내용	작성일자	작성자
-----	-----	-----	------	-----

2-2) 약한 개체가 참여하는 일대다 관계는 외래키를 포함해서 기본키로 지정.

- 포함 : 회원(강한개체)와 학생(약한개체)의 관계

회원의 기본키인 school_id를 학생의 외래키로 지정하고 외래키를 포함하여 기본키를 지정한다.

이때 school_id는 학번과 같은 뜻이므로 속성명을 학번으로 바꾸고 이를 외래키이자 기본키로 지정한다.

→ 학생 릴레이션

기본키 : 학번, 외래키 : 학번

학번	이름	학과
----	----	----

- 포함 : 회원(강한개체)와 교수(약한개체)의 관계

회원의 기본키인 school_id를 교수의 외래키로 지정하고 외래키를 포함하여 기본키를 지정한다.

이때 school_id는 사번과 같은 뜻이므로 속성명을 사번으로 바꾸고 이를 외래키이자 기본키로

지정한다.

→ 교수 릴레이션

기본키 : 사번, 외래키 : 사번

사번	이름
----	----

- 포함 : 과목(강한개체)와 강의(약한개체)의 관계

과목의 기본키인 과목코드를 강의의 외래키로 지정하고 외래키를 포함하여 기본키를 지정한다.

→ 강의 릴레이션

기본키 : {과목코드, 주차, 차시}, 외래키 : 과목코드

과목코드	주차	차시	강의내용
------	----	----	------

- 포함 : 과목(강한개체)와 과제(약한개체)의 관계

과목의 기본키인 과목코드를 과제의 외래키로 지정하고 외래키를 포함하여 기본키를 지정한다.

→ 과제 릴레이션

기본키 : {과목코드, 과제코드}, 외래키 : 과목코드

과목코드	과제코드	시작일	마감일	과제내용
------	------	-----	-----	------

규칙 3 : 다대다 관계는 릴레이션으로 변환한다. → 수강신청, 수강, 수행

- 수강신청 릴레이션 : 학생(기본키 : 학번)과 과목(기본키 : 과목코드)의 관계,

→ 과목코드, 학번 : 외래키

→ {과목코드, 학번} : 기본키

학번	과목코드
----	------

- 수행 릴레이션 : 학생(기본키 : 학번)과 과제(기본키 : {과목코드, 과제코드})의 관계,

→ {과목코드, 과제코드}, 학번 : 외래키,

→ {과목코드, 과제코드, 학번} : 기본키

학번	과목코드	과제코드	수행여부
----	------	------	------

- 수강 릴레이션

학생(기본키 : 학번)과 강의(기본키 : {과목코드, 주차, 차시})의 관계,

→ {과목코드, 주차, 차시}, 학번 : 외래키,

→ {과목코드, 주차, 차시, 학번} : 기본키

학번	과목코드	주차	차시	강의여부
----	------	----	----	------

→ 논리적 설계 릴레이션 스키마

- 회원 릴레이션

기본키 : school_id

School_id	아이디	비밀번호	가입날짜
-----------	-----	------	------

- 학생 릴레이션

기본키 : 학번, 외래키 : 학번

학번	이름	학과
----	----	----

- 교수 릴레이션

기본키 : 사번, 외래키 : 사번

사번	이름
----	----

- 과목 릴레이션

기본키 : 과목코드, 외래키 : 교수사번

과목코드	과목명	교수명	수강료	교수사번
------	-----	-----	-----	------

- 수강신청 릴레이션

→ 과목코드, 학번 : 외래키

→ {과목코드, 학번} : 기본키

학번	과목코드
----	------

- 강의 릴레이션

기본키 : {과목코드, 주차, 차시}, 외래키 : 과목코드

과목코드	주차	차시	강의내용
------	----	----	------

- 과제 릴레이션

기본키 : {과목코드, 과제코드}, 외래키 : 과목코드

과목코드	과제코드	시작일	마감일	과제내용
------	------	-----	-----	------

- 수강 릴레이션

→ {과목코드, 주차, 차시}, 학번 : 외래키,

→ {과목코드, 주차, 차시, 학번} : 기본키

학번	과목코드	주차	차시	강의여부
----	------	----	----	------

- 수행 릴레이션

→ {과목코드, 과제코드}, 학번 : 외래키,

→ {과목코드, 과제코드, 학번} : 기본키

학번	과목코드	과제코드	수행여부
----	------	------	------

- 강의자료실 릴레이션

기본키 : 글번호, 외래키 : 작성자

글번호	글제목	글내용	작성일자	작성자
-----	-----	-----	------	-----

3-2. 테이블 명세서

- 회원

아이디 또한 null값이 올 수 없고 중복되면 안되므로 이는 대체키(AKI)로 지정한다.

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
School_id	Varchar(20)	N		PK		
아이디	Varchar(20)	N				
비밀번호	Varchar(20)	N				
가입날짜	date	Y				

- 학생

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
학번	Varchar(20)	N		PK	FK	P로 시작하지 않는다.
이름	Varchar(20)	N				
학과	Varchar(20)	Y				

- 교수

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
사번	Varchar(20)	N		PK	FK	P로 시작한다.
이름	Varchar(20)	N				

- 과목

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
과목코드	Varchar(20)	N		PK		

과목명	Varchar(20)	N				
교수명	Varchar(20)	N				
수강료	int	N	0원			
교수사번	Varchar(20)	N			FK	

- 수강신청

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
학번	Varchar(20)	N		PK	FK	
과목코드	Varchar(20)	N		PK	FK	

- 강의

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
과목코드	Varchar(20)	N		PK	FK	
주차	Int	N		PK		
차시	Int	N		PK		
강의내용	Varchar(40)	Y				

- 과제

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
과목코드	Varchar(20)	N		PK	FK	
과제코드	Varchar(20)	N		PK		
시작일	Date	N				
마감일	Date	N				
과제내용	Varchar(40)	Y				

- 수강

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
과목코드	Varchar(20)	N		PK	FK	
주차	INT	N		PK	FK	
차시	INT	N		PK	FK	
학번	Varchar(20)	N		PK	FK	
수강여부	Varchar(10)	N				수강, 지각, 결석, 미결

- 수행

속성이름	데이터 타입	널 허용 여부	기본값	기본키	외래키	제약조건
과목코드	Varchar(20)	N		PK	FK	
과제코드	Varchar(20)	N		PK	FK	
학번	Varchar(20)	N		PK	FK	

4. 물리적 모델

4-1. 개체의 이름과 속성의 이름을 실제 사용할 테이블 이름과 컬럼 이름으로 변경해준다.

개체	회원		학생	
구분	Logical	Physical	Logical	physical
이름	회원	member	학생	student
기본키	school_id	school_id	학번	s_id
속성	아이디 비밀번호 가입날짜	id pw register_date	이름 학과	s_name major

개체	교수		과목	
구분	Logical	Physical	Logical	physical
이름	교수	professor	과목	class
기본키	사번	p_id	과목코드	c_id
속성	이름	p_name	과목명 교수명 수강료 교수사번	c_name p_name tuition_fee p_id

개체	과제		강의	
구분	Logical	Physical	Logical	physical
이름	과제	hw	강의	lectue
기본키	과목코드 과제코드	c_id h_id	과목코드 주차 차시	c_id week time
속성	시작일 마감일 과제내용	start end h_content	강의내용	l_content

개체	강의자료실		수강신청	
구분	Logical	Physical	Logical	physical
이름	강의자료실	lecture_resource	수강신청	regist
기본키	글번호	w_id	과목코드 학번	c_id s_id

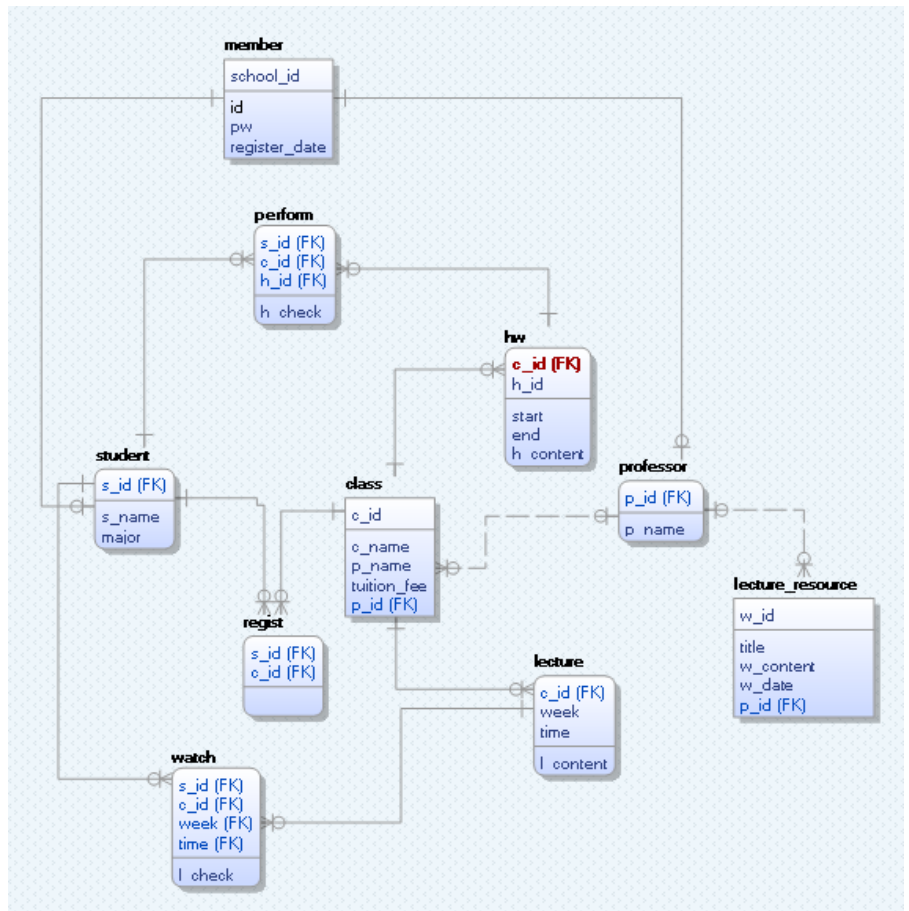
속성	글제목 글내용 작성일자 작성자	title w_content w_date p_id		
----	---------------------------	--------------------------------------	--	--

개체	수강		수행	
구분	Logical	Physical	Logical	physical
이름	수강	watch	수행	perform
기본키	과목코드 주차 차시 학번	c_id week time s_id	과목코드 과제코드 학번	c_id h_id s_id
속성	수강여부	l_check	수행여부	h_check

4-2. 데이터 타입

회원(member)		학생(student)		교수(professor)	
속성이름	데이터 타입	속성이름	데이터 타입	속성이름	데이터 타입
school_id	Varchar(20)	s_id	Varchar(20)	p_id	Varchar(20)
id	Varchar(20)	s_name	Varchar(20)	p_name	Varchar(20)
pw	Varchar(20)	major	Varchar(20)		
register_date	date				
과목(class)		수강신청(regist)		강의(lecture)	
속성이름	데이터 타입	속성이름	데이터 타입	속성이름	데이터 타입
c_id	Varchar(20)	s_id	Varchar(20)	c_id	Varchar(20)
c_name	Varchar(20)	c_id	Varchar(20)	week	Int
p_name	Varchar(20)			time	Int
tuition_fee	int			l_content	Varchar(40)
p_id	Varchar(20)				
과제(hw)		수강(watch)		수행(perform)	
속성이름	데이터 타입	속성이름	데이터 타입	속성이름	데이터 타입
c_id	Varchar(20)	c_id	Varchar(20)	c_id	Varchar(20)
h_id	Varchar(20)	week	Int	h_id	Varchar(20)
start	date	time	Int	s_id	Varchar(20)
end	Date	s_id	Varchar(20)	h_check	Varchar(10)
h_content	Varchar(40)	l_check	Varchar(10)		

강의자료실 (lecture_resource)	
속성이름	데이터 타입
w_id	Int
title	Varchar(20)
w_content	Varchar(100)
w_date	Date
p_id	Varchar(20)



5. MySQL에 DBMS 만들기

- 생성된 DBMS

1. 회원

	school_id	id	pw	register_date
▶	20192643	sjs2000	sjs130	2020-03-01
	20192644	tjwlstmf	sjs130	2020-03-01
	20192645	tjwlstmf123	sjs130	2020-03-01
	20192646	adflkadrk	sjs130	2020-03-01
	p01	sjsjsj	sjs130	2020-03-01
	p02	sj	sjs130	2020-03-01
	p03	kkk	sjs130	2020-03-01
	p04	dasf	sjs130	2020-03-01
*	NULL	NULL	NULL	NULL

2. 학생

	s_id	s_name	major
▶	20192643	서진솔	응통
	20192644	서진	응통
	20192645	서	응통
	20192646	서진수	NULL
*	NULL	NULL	NULL

3. 교수

	p_id	p_name
▶	p01	김승태
	p02	김명호
	p03	오영석
	p04	김영화
*	NULL	NULL

4. 과목리스트 (class)

	c_id	c_name	p_name	tuition_fee	p_id
▶	c01	자료구조	김승태	30000	p01
	c02	데배설	오영석	40000	p03
	c03	프로그래밍	김명호	50000	p02
	c04	기통	김영화	50000	p04
*	NULL	NULL	NULL	NULL	NULL

5. 각 학생의 수강 과목 리스트 (regist)

	s_id	c_id
▶	20192643	c01
	20192643	c02
	20192644	c01
	20192644	c02
	20192645	c01
	20192645	c02
	20192645	c03
*	NULL	NULL

6. 과목별 주차 및 차시 강의 정보 (lecture)

	c_id	week	time	l_content
▶	c01	14	1	자료구조란?
	c01	14	2	링크드리스트
	c01	15	1	이진트리
	c02	14	1	SQL 이란
	c02	14	2	SQL 실습
	c03	14	1	C언어란
	c03	14	2	C언어 실습
	c04	14	1	통계란
	c04	14	2	가설검정
*	NULL	NULL	NULL	NULL

7. 과목별 과제 정보 (hw)

	c_id	h_id	start	end	h_content
▶	c01	h01	2021-12-01	2021-12-10	자료구조 과제1
	c01	h02	2021-12-11	2021-12-30	자료구조 과제2
	c02	h01	2021-12-01	2021-12-10	데베설 과제1
	c02	h02	2021-12-11	2021-12-30	데베설 과제2
★	NULL	NULL	NULL	NULL	NULL

8. 수강기록 (수강 및 미수강 여부 포함) (watch)

	s_id	c_id	week	time	l_check
▶	20192643	c01	14	1	수강
	20192643	c01	14	2	수강
	20192644	c01	14	1	미수강
	20192645	c02	14	1	수강
	20192645	c03	14	1	결석
★	NULL	NULL	NULL	NULL	NULL

9. 과제기록 (제출 및 미제출 여부 포함) (perform)

	s_id	c_id	h_id	h_check
▶	20192643	c01	h01	제출
	20192644	c01	h01	제출
	20192645	c01	h02	미제출
	20192643	c02	h01	지각제출
	20192644	c02	h02	지각제출
★	NULL	NULL	NULL	NULL

10. 강의자료실

	w_id	title	w_content	w_date	p_id
▶	1	오늘 수업자료	자료구조 자료	2021-12-10	p01
	2	내일 수업자료	연결리스트 자료	2021-12-10	p01
	3	데베설 수업자료	NULL	2021-12-10	p02
★	NULL	NULL	NULL	NULL	NULL