

# Project#1 수강신청 사이트

☰ 수업일	
📌 유형	실습

## Database Schema

### 주어진 스키마

1차 설계

2차 설계

## ? Query

로그인

수강편람

희망수업

수강신청

수강취소

본인 정보 변경

시간표 생성

설강 및 폐강

학생 정보 조회 및 수정

통계조회 (OLAP)

전교생 시간표 조회

수강신청 기간 종료

## Plan

### 공통

1. 로그인

2. 로그아웃

3. 수강편람

### 학생

1. 수강신청

2. 장바구니

3. 내 정보

### 관리자

설강 및 폐강

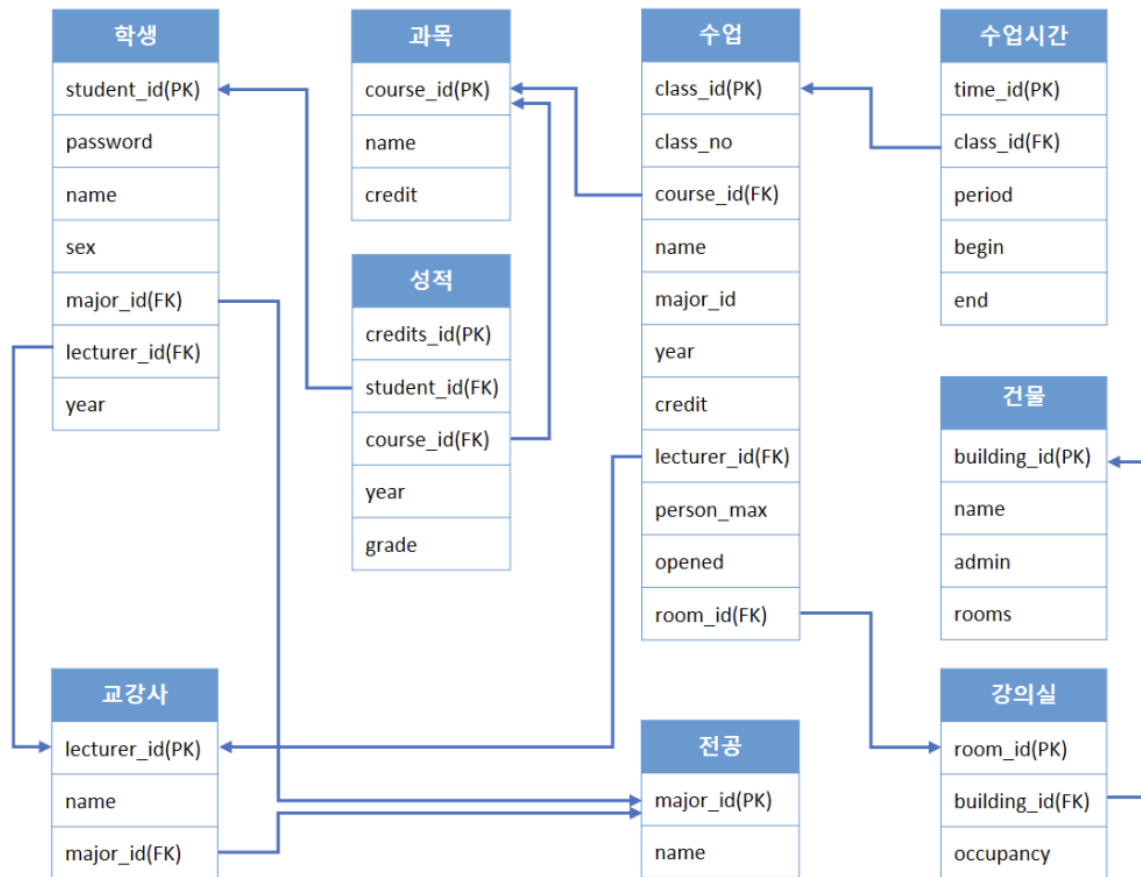
학생 정보 조회

통계조회

학생 시간표 조회

## Database Schema

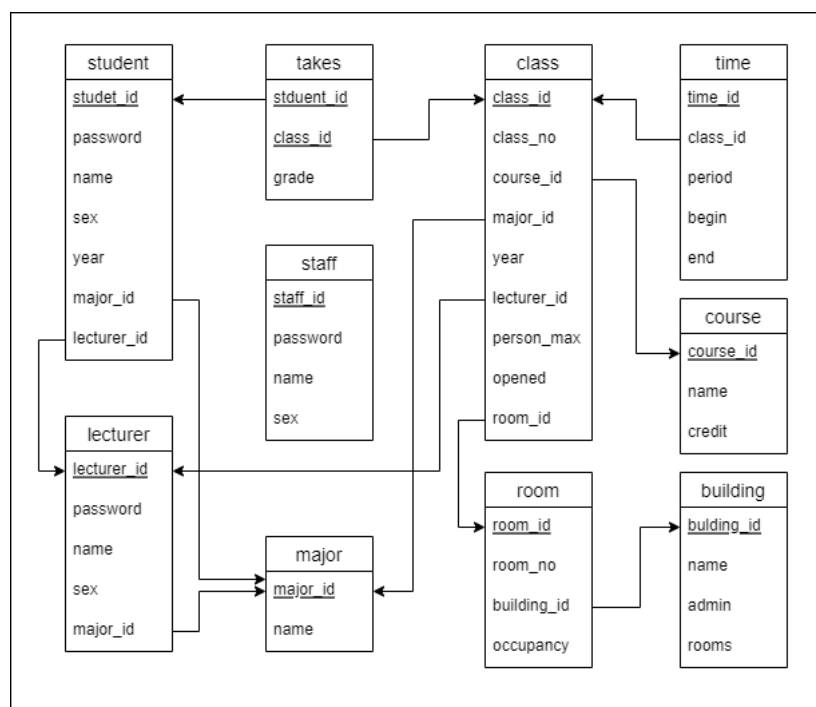
### 주어진 스키마



staff(관리자) 교수 추가했을 때 구분 할 것인

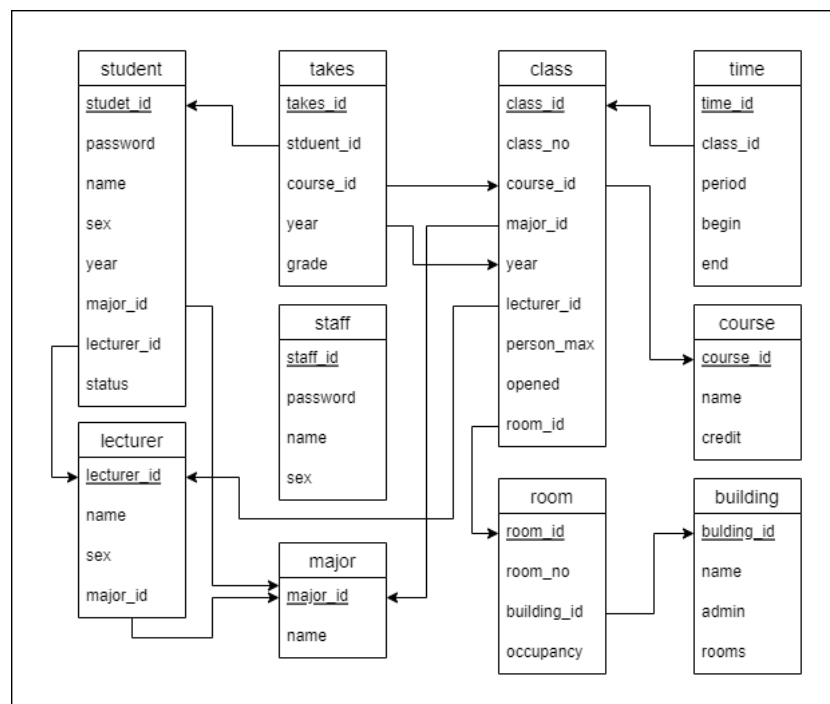
takes: class\_id로 할 지 course\_id 로 할 지

## 1차 설계



- staff(관리자)
  - 관리자의 정보를 저장해주는 릴레이션이다.
- building(건물)
- room(강의실)
  - room\_id를 room의 구분하는 용도(PK)이고 room\_no는 강의실을 알려주는 용으로 새로 추가하였다.
  - 기존의 방식은 모든 강의실이 겹치는 번호가 없다는 설정인 것 같았지만 그러는 것보다는 새로운 속성을 추가하는 것이 좋다고 생각하였다.
  - ex) IB/BT관 207호 인 경우 IB/BT에 대한 building\_id 와 room\_no(=207), room\_id(구분을 위한 임의의 중복이 안되는 숫자), occupancy(이 강의실이 수용할 수 있는 최대 인원수)가 하나의 행이다.
- course(과목)
- major(전공)
- takes(수강과목)
  - 현재까지 수강한 과목과 그 성적, 수강중인 과목, 그리고 수강희망과목을 알려준다.
  - 수업을 가져오기 위해 class\_id를 사용하는 것으로 설정하였다.
  - grade의 속성이 N 이면 아직 성적이 안나온 것(수강중)이고, W는 희망수업을 나타낸다.
- lecturer(교강사)
  - password, sex 속성을 추가하였다
- student(학생)
- class(수업)
  - name과 credit은 course를 join함으로서 알 수 있다고 생각(중복)되어서 삭제하였다.
  - major\_id를 FK로 설정했다.

## 2차 설계



- staff에 교수님들을 아예 추가하는 방식으로 바꾸었다.
  - 원래는 관리자 로그인을 하면 staff와 lecturer table을 부르려 했지만 staff 하나만 부르고 교수님은 모두 관리자가 되는 것으로 설정하였다.

- 동시에 lecturer 에서 password는 지웠다.
- takes 에서 course\_id 와 year을 class\_id로 바꾸려 했지만 년도별 성적이 필요할 수도 있겠다 생각되어 다시 원래 상태로 돌렸다
  - 여기서 (course\_id, year) 을 FK로 사용하려 했지만 오류가 발생했다.
    - Error Code: 1822. Failed to add the foreign key constraint. Missing index for constraint 'takes\_ibfk\_2' in the referenced table 'class'
  - 일단 course\_id 만 FK로 설정하고 진행하였다.
- student에 status를 추가하였다
  - 재학/휴학/졸업 등을 표현할 수 있다.
- 문제점
  - takes에서 많은 것을 처리하려다보니 비효율적인 것 같다
    - 시간표를 만드려고 했더니 course\_id 와 year로는 어떤 것을 들었는 지 알 수 없다
    - 수강신청 시점에서는 applied 테이블을 따로 만들고 학기가 끝나면 takes(credits)로 넘기는 방향으로 바꾸어야겠다.

## ? Query

### 공통

### 로그인

- 학생 로그인

```
select * from student where student_id=? and password=?
```

- 관리자 로그인

```
select * from staff where staff_id=? and password=?;
```

### 수강편람

```
with
cls_crs(class_id, class_no, course_id, major_id, year, lecturer_id, person_max, opened, room_id, class_name, credit) as (
  select class_id, class_no, course_id, major_id, year, lecturer_id, person_max, opened, room_id, name, credit
    from class
   natural join course
), # (0) class와 course를 course_id로 묶음
class_ins(class_id, course_id, ins_name, person_max, room_id, class_name) as (
  select C.class_id, C.course_id, L.name, C.person_max, C.room_id , C.class_name
    from cls_crs as C
   join lecturer as L
    on C.lecturer_id=L.lecturer_id
), # (1) clas_crs와 lecturer의 lecturer_id 가 같은 애들
room_building(building_id, room_id, build_name) as (
  select building_id, room_id, name
    from room R
   natural join building B
), # (2) room과 building의 building_id가 같은 애들
CLBR(room_id, class_id, course_id, ins_name, person_max, class_name, building_id, build_name) as (
  select *
    from class_ins as CI
   natural join room_building as RB
), # (3) (1),(2)를 room_id로 묶어둠
totTime(class_id, begin1, end1, begin2, end2) as (
  select t1.class_id, t1.begin, t1.end, t2.begin, t2.end
    from time as t1
   left join time as t2
    on t1.period < t2.period and t1.class_id=t2.class_id
   where t1.period=1
), # (4) self join을 통해 하나의 행에 모든 수업시간이 들어가게 함
# t1.period < t2.period 를 통해 중복을 없앴음
total as (
  select CLBR.class_id, course_id, class_name, ins_name, person_max, building_id, build_name, begin1, end1, begin2, end2
    from CLBR
   natural left join totTime
```

```
)# (5) (3),(4)를 class_id로 합치면 준비완료
select * from total [조건문] order by class_name;
```

### [조건문]

- 수업번호, 학수번호, 교과목명, 개설년도를 입력받아 그에 맞는 쿼리를 추가해준다.
  - 데이터를 분석해 보았을 때 년도별로 수강편람을 확인하는 기능까지 추가해줘야 한다고 생각하였다. (2022년 수강신청이라고 가정한 상황이라고 추측하였다)

## 학생

### 희망수업

```
insert into takes values([id], [stu_id], [crs_id], [year], "W");
```

- [id] 자동으로 증가되는 값이다
- [stu\_id] 희망수업신청을 하는 학생의 아이디
- [crs\_id] 학생이 희망하는 수업의 아이디
- [year] 현재 년도
- grade → "W" : Wanted를 의미
  - 수강신청 기간이 끝나면 지워진다.

### 수강신청

#### 1. 희망수업에 등록한 경우

##### a. 조건

- i. 이전 성적이 B0 이상인 경우  
= 아래 sql 을 시행하여 얻은 모든 값이 A+, A0, B+, B0가 아닐 때여야함

```
select grade from takes where student_id="[stu_id]" and course_id="[crs_id]";
```

- ii. 정원이 다 찼을 경우  
= 정원에 대한 새로운 속성이 필요할 것 같다.
- iii. 동일 시간대에 신청한 경우  
= 현재 수강신청한 과목들의 시간과 현재 과목을 비교하여 처리한다.
- iv. 최대 학점(18)을 넘기는 경우  
= 아래 수행 결과가 18을 넘기는지 확인

```
select sum(credit)
from takes
natural join course
where student_id="[stu_id]" and year="[year]";
```

- b. 성공: grade → "N" : 결정되지 않았음(None). 수업을 수강중임을 나타냄

```
update takes set grade="N" where student_id="[stu_id]";
```

#### 2. 그렇지 않는 경우

##### a. 성공

```
insert into takes values([id], [stu_id], [crs_id], [year], "N");
```

## 수강취소

```
delete from takes where couse_id="[crs_id]" and student_id="[stu_id]";
```

## 본인 정보 변경

비밀번호 변경

```
update student set password="[newPwd]" where student_id="[stu_id]";
```

## 시간표 생성

스키마를 잘못짜서 시간을 가져올 수 없다.

applied(student\_id, class\_id) 와 같은 테이블을 따로 만들어서 관리해야 할 것 같다.

## 관리자

### 설강 및 폐강

#### 설강

1. 새로운 강의인 경우

```
insert into course values ("[crs_id]", "[name]", "[credit]");  
insert into class values ("[cls_id]", "[cls_no]", "[crs_id]", "[mj_id]", "[year]", "[lec_id]", "[person_max]", "[opened]", "[room_
```

2. 그렇지 않은 경우

```
insert into class values ("[cls_id]", "[cls_no]", "[crs_id]", "[mj_id]", "[year]", "[lec_id]", "[person_max]", "[opened]", "[room_
```

#### 폐강

1. 똑같은 course가 존재 하지 않는다면
  - a. 이건 primary key 설정을 통해 자동으로 지울 수도 있을 것 같음 (알아보기)

```
delete from class where class_id="[cls_id]";  
delete from course where course_id="[crs_id]";
```

2. 그렇지 않은 경우

```
delete from class where class_id="[cls_id]";
```

## 학생 정보 조회 및 수정

학생 기본 정보(학생ID, 이름, 성별, 학년, 지도교수ID, 지도교수 이름, 전공)

```
select S.student_id as stduent_id, S.name as student_name, S.sex as sex, S.year as year, S.lecturer_id as lecturer_id, L.name as lectu  
from student S  
join lecturer L  
on S.lecturer_id=L.lecturer_id  
join major M  
on S.major_id=M.major_id  
where student_id="[stu_id]";
```

#### 학생 수강 정보

```
select * from takes where student_id="[stu_id]";
```

수정은 학생과 비슷하기에 생략.

## 통계조회 (OLAP)

학점 평균 구하기

1. 한 년도(학기)의 학점을 가져온다

```
select grade from takes where student_id="[stu_id]" and year="[year]";
```

2. 그 년도의 총 학점을 구한다

```
select sum(credit)
from takes
natural join course
where student_id="[stu_id]" and year="[year]"
```

- 학점을 ABC 형식이 아닌 4.5/4.0/3.5 ... 로 표현하면 더 좋을 수도 있겠다는 생각이 들었다
  - 아니면 `translate(char, float)` 느낌의 테이블을 만들어도 될거같다. (애초에 실수로 표현하는게 좋을 것 같도
3. 그 년도 과목마다 평균을 빼준 값을 구한다. (+ order by)

```
select grade-avg as diff
from takes
where student_id="[stu_id]" and year="[year]"
order by diff
limit 10;
```

## 전교생 시간표 조회

마찬가지로 지금 스키마로는 알아낼 수 없다. 스키마를 다시 만들어야할 것 같다.

## 수강신청 기간 종료

== 희망 수업 삭제

take의 grade=="W" 인 행을 모두 지운다

```
delete from takes where grade="W";
```

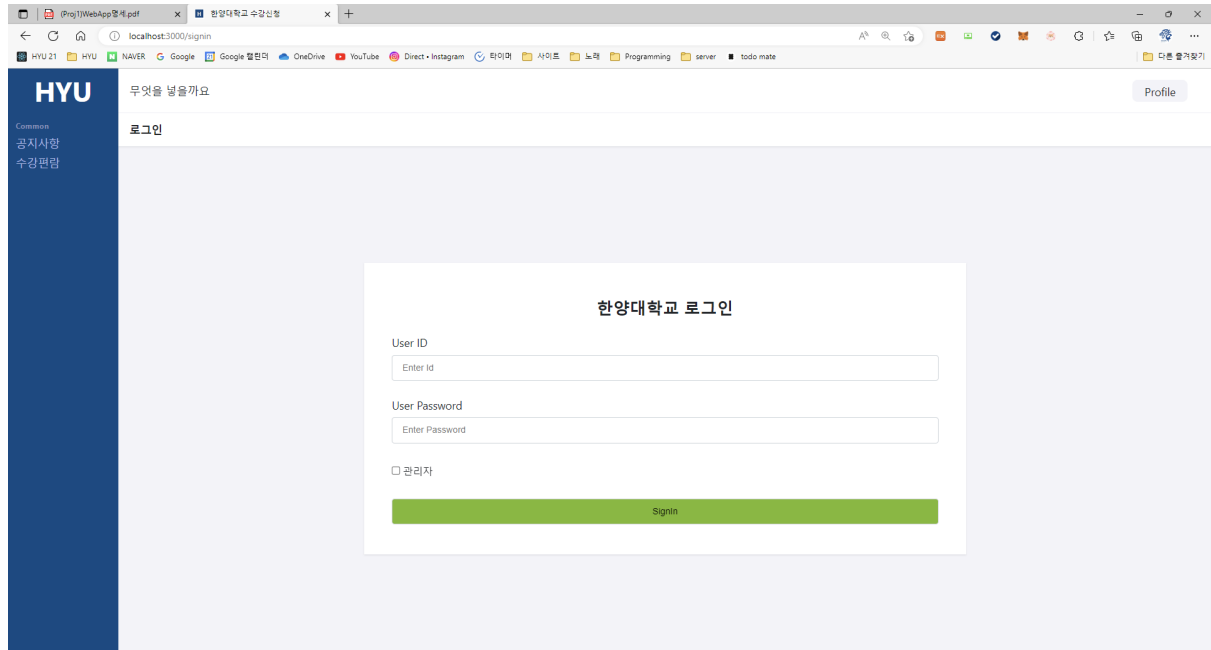


## Plan

## 공통

### 1. 로그인

디자인



## 동작

- 관리자로 로그인할 것인지 확인한다
  - 관리자는 lecturer + staff 이다.
  - 학생은 관리자로 로그인할 수 없고 관리자는 일반 로그인을 할 수 없다.
- 아이디(학번)과 비밀번호를 입력 받는다.
- 입력받은 값과 비교를 해 인증되면 공지사항 페이지로 이동시킨다.
  - 이때 로그인 정보를 sessionStorage를 이용해 저장한다.

## 기타

- 이미 로그인한 경우 프로필 페이지로 이동시킨다.
- 입력값이 부족하면 경고창을 띄운다.

## 2. 로그아웃

### 동작

- 세션값을 지우고 로그아웃시킨다.
- 공지사항 페이지로 이동시킨다.

### 기타

- 로그인하지 않은 경우 로그인 페이지로 보낸다

## 3. 수강편람

### 디자인



<div>HYU</div> <div>Common</div> <div>공지사항</div> <div>수강편람</div>	무엇을 넣을까요							Profile
	수강편람							
	<input type="text" value="수업번호"/>	<input type="text" value="학수번호"/>	<input type="text" value="교과목명"/>	<input type="text" value="2022"/>	<input type="button" value="검색"/>	<input type="button" value="초기화"/>		
	수업번호	학수번호	교과목명	교강사	수업시간	수강/정원	강의실	년도
	10005	CIE3033	건설경제학	서종원	목요일 11:00 - 12:30 금요일 10:30 - 12:00	25	제 1공학관 212	2022
	10073	ARE3069	건설사업관리1	유기찬	화요일 13:00 - 14:30 목요일 09:00 - 10:30	30	제 1법학관 407	2022
	10013	CIE4068	건설환경공학종합설계2	유재석	토요일 09:00 - 11:00	39	제 1공학관 212	2022
	10075	ARE4020	건축공학실험	이충선	화요일 13:00 - 15:00 화요일 15:00 - 17:00	18	제 2공학관 211	2022
	10074	ARE4014	건축구조계획	김효진	금요일 13:00 - 14:30 금요일 14:30 - 16:00	25	제 2공학관 211	2022
	10072	ARE3009	건축구조역학2	최창식	화요일 16:00 - 17:30 목요일 10:30 - 12:00	17	IT / BT 305	2022
	10076	ARE4022	건축음향	전진용	수요일 16:00 - 17:30 금요일 10:30 - 12:00	30	제 2공학관 211	2022
	10079	ARE5013	건축종합설계2	박운중	월요일 13:00 - 14:00 월요일 14:00 - 18:00	30	IT / BT 305	2022
	10140	TXE4065	고분자구조	임승순	월요일 09:00 - 10:30 화요일 10:30 - 12:00	30	IT / BT 305	2022
	10142	CHM4013	고분자물성	강용수	월요일 13:00 - 14:30 수요일 09:00 - 10:30	34	제 1공학관 212	2022
	10094	CHM4013	고분자물성	서경도	화요일 10:30 - 12:00 목요일 10:30 - 16:00	24	자연과학관 507	2022
	10125	ICH4022	고분자재료	김성훈	월요일 14:30 - 16:00 수요일 09:00 - 10:30	30	제 2공학관 211	2022
					월요일 13:00 - 14:30			

## 학생

### 1. 수강신청

#### 디자인

HYU

Common

공지사항

수강편람

Student

수강신청

장바구니

내 정보

무엇을 넣을까요

수강신청

수업번호

학수번호

교과목명

2022

검색

초기화

수강신청	수업번호	학수번호	교과목명	교강사	수업시간	수강/정원	강의실	년도
신청	10005	CIE3033	건설경제학	서종원	목요일 11:00 - 12:30 금요일 10:30 - 12:00	25	제 1공학관 212	2022
신청	10073	ARE3069	건설사업관리1	유기찬	화요일 13:00 - 14:30 목요일 09:00 - 10:30	30	제 1법학관 407	2022
신청	10013	CIE4068	건설환경공학종합설계2	유재석	토요일 09:00 - 11:00	39	제 1공학관 212	2022
신청	10075	ARE4020	건축공학실험	이충선	화요일 13:00 - 15:00 화요일 15:00 - 17:00	18	제 2공학관 211	2022
신청	10074	ARE4014	건축구조계획	김효진	금요일 13:00 - 14:30 금요일 14:30 - 16:00	25	제 2공학관 211	2022
신청	10072	ARE3009	건축구조역학2	최창식	화요일 16:00 - 17:30 목요일 10:30 - 12:00	17	IT / BT 305	2022
신청	10076	ARE4022	건축음향	전진용	수요일 16:00 - 17:30 금요일 10:30 - 12:00	30	제 2공학관 211	2022
신청	10079	ARE5013	건축종합설계2	박운중	월요일 13:00 - 14:00 월요일 14:00 - 18:00	30	IT / BT 305	2022
신청	10140	TXE4065	고분자구조	임승순	월요일 09:00 - 10:30 화요일 10:30 - 12:00	30	IT / BT 305	2022
신청	10142	CHM4013	고분자물성	강용수	월요일 13:00 - 14:30 수요일 09:00 - 10:30	34	제 1공학관 212	2022
신청	10094	CHM4013	고분자물성	서경도	화요일 10:30 - 12:00 목요일 10:30 - 12:00	24	자연과학관 507	2022
신청	10125	ICH4022	고분자재료	김성훈	월요일 14:30 - 16:00 수요일 09:00 - 10:30	30	제 2공학관 211	2022

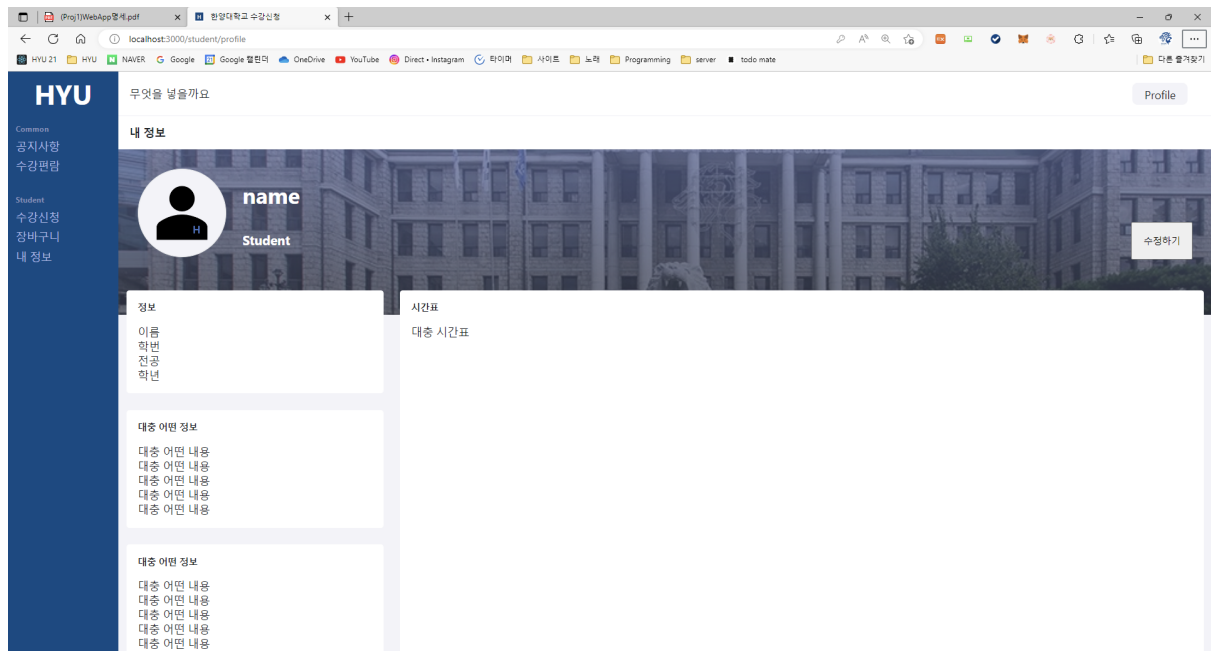
수업번호, 학수번호, 교과목명을 입력하면 알아낼 수 있다.

### 2. 장바구니

수강신청 페이지에서 원하는 과목을 선택하면 장바구니 페이지에서 따로 확인할 수 있다.(그림에는 나와있지 않음)

### 3. 내 정보

#### 디자인



## 관리자

### 설강 및 폐강

설강 또는 폐강할 과목의 필요한 데이터들을 입력 받아 sql문을 실행한다.

### 학생 정보 조회

조회할 학생의 ID를 받아온다.

### 통계조회

궁금한 통계 데이터를 선택한다.

### 학생 시간표 조회

궁금한 학생의 ID를 받아온다.