웹사이트 개발 순서

1.포트폴리오를 위한 주제 선정

쇼핑몰 - 로그인, 상품 소개, 구매, 장바구니, 좋아요, 구매내역 확인, 상품수령 확인, 구매 후기, 문의

패스트푸드 - 로그인, 상품 소개, 구매, 장바구니, 좋아요, 구매내역 확인, 구매 후기, 문의

기업 - 로그인, 기업 소개 및 공지, 부서 분류, 부서별 게시판, 메시지 및 파일첨부

웨이트 Gym - 로그인, 공지, 체육관 소개, 트레이너 정보, 운동 정보, 문의, 등록 기간 확인, 회원 게시판

2. 요구 분석 명세서 작성

ID - RQ-001

요구사항명 - ID 글자수 제한

요구사항 내용 - 회원가입시 ID는 최대 12자리로 제한한다

ID - RQ-002

요구사항명 - ID 형식 제한

요구사항 내용 - ID에는 특수문자가 들어갈 수 없다

3. 스토리보드 작성

4. DB설계

- ERD 작성(ppt) -> 개념모델링

- 논리적 설계, 물리적 설계(엑셀)

- 데이터베이스 구축(sql 파일)

- 기능 테스트 쿼리문

5. 웹사이트 구현

문의 게시판 보기

SELECT \* FROM board where board\_category\_code = 1;

자유 게시판 보기

SELECT \* FROM board where board\_category\_code = 2;

공지 게시판 보기

SELECT \* FROM board where board\_category\_code = 3;

운동정보 게시판 보기

SELECT \* FROM board where board\_category\_code = 4;

체육관 소개 게시판 보기

SELECT \* FROM board where board\_category\_code = 5;

뒷담 게시판 보기

SELECT \* FROM board where board\_category\_code = 9;

인증 게시판 보기

SELECT \* FROM board where board\_category\_code = 10;

해당 시간에 진행중인 pt 일정이 있는지 검색

SELECT \* FROM pt where pt\_schedule = '13:00~15:00';

현재 진행중인 PT 정보 검색

select member.member\_name, pt\_schedule, pt\_round, trainer\_schedule, member\_address, member\_phone

from pt

join trainer

on pt\_trainer\_no = trainer.trainer\_no

join member

on trainer.trainer\_member\_id = member.member\_id;

게시글 작성

insert into board(board\_category\_code, board\_writer, board\_title, board\_contents)

values('int', ' ', ' ',' ');

회원정보 검색

SELECT member\_name,

member\_nickname,

member\_address,

member\_phone,

member\_period,

member\_course\_no

FROM member

where member\_id = 'avbnn';

A회원의 pt정보와 담당 trainer 정보 검색

SELECT member.member\_name,

member.member\_phone,

pt.pt\_schedule,

pt.pt\_round,

pt.pt\_contents,

detail.member\_name trainer\_name,

detail.member\_phone trainer\_phone

FROM pt

join member

on pt.pt\_member\_id = member.member\_id

join (select \* from trainer join member on trainer.trainer\_member\_id = member.member\_id) as detail

on pt.pt\_trainer\_no = detail.trainer\_no

where pt.pt\_member\_id = 'lee2324';

트레이너 정보 검색

SELECT member\_name,

member\_phone,

trainer\_schedule,

member\_address,

member\_period

FROM trainer

join member

on trainer\_member\_id = member.member\_id;

게시글의 첨부파일 확인

SELECT \* FROM upload where upload\_board\_no = 3;

글 내용에 ‘파인 옷’이 들어가는 글을 검색

SELECT \* FROM board where board\_contents like '%파인 옷%';

글 제목에 ‘파인 옷’이 들어가는 글을 검색

SELECT \* FROM board where board\_title like '%파인 옷%';

작성자명이 ‘lee’로 시작하는 글을 검색

SELECT \* FROM board where board\_writer like 'lee%';

2019-05-03 00:00:00 이후 작성된 글을 검색

SELECT \* FROM board where board\_register\_date like '2019-05-03%';

수강 등록 후 member테이블의 course\_no와 program 테이블의 program\_total을 수정

insert into course(course\_program\_no, course\_member\_id, course\_period)

values(1, 'avbnn', '2019-04-01~2019-05-31');

update member

set member\_course\_no = (select course\_no from course where course\_member\_id = 'avbnn')

where member\_id = 'avbnn';

update program

set program\_total = program\_total+1

where program\_no = 1;

A회원의 수강정보를 검색

SELECT member.member\_name as '이름',

course.course\_no as '수강번호',

detail.member\_name as '담당 트레이너 이름',

course\_period as '수강 기간',

program.program\_title as '프로그램명',

program.program\_contents as '프로그램 내용',

program.program\_schedule as '프로그램 일정',

program.program\_total\_limit as '최대 인원',

program.program\_total as '현재 인원'

FROM member

join course

on member.member\_course\_no = course.course\_no

join program

on course.course\_program\_no = program.program\_no

join (select \* from trainer join member on trainer.trainer\_member\_id = member.member\_id) as detail

on program.program\_trainer\_no = detail.trainer\_no

where member.member\_id = 'aaafgeg';

A회원 정보를 검색하는 프로시저로 변환

drop procedure if exists search\_member\_program;

delimiter //

create procedure search\_member\_program( in in\_member\_id varchar(45) )

begin

SELECT member.member\_name as '이름',

course.course\_no as '수강번호',

detail.member\_name as '담당 트레이너 이름',

course\_period as '수강 기간',

program.program\_title as '프로그램명',

program.program\_contents as '프로그램 내용',

program.program\_schedule as '프로그램 일정',

program.program\_total\_limit as '최대 인원',

program.program\_total as '현재 인원’

FROM member

join course

on member.member\_course\_no = course.course\_no

join program

on course.course\_program\_no = program.program\_no

join (select \* from trainer join member on trainer.trainer\_member\_id = member.member\_id) as detail

on program.program\_trainer\_no = detail.trainer\_no

where member.member\_id = in\_member\_id;

end //

delimiter ;

show procedure status;

Gym 트리거 생성

course 테이블에 insert시

-> 1.프로그램 no가 일치하는 프로그램 테이블의 program\_total을 증가시킨다

-> 2. 멤버 id가 일치하는 멤버 테이블의 member\_course\_no에 course\_no을 입력한다

use gym;

drop trigger if exists add\_course;

delimiter //

create trigger add\_course after insert on course

for each row

begin

declare \_program\_total int default 0;

declare \_member\_course\_no int default 0;

set \_program\_total = (select count(\*) from course where new.course\_program\_no = course\_program\_no);

set \_member\_course\_no = new.course\_no;

update program

set program\_total = \_program\_total

where program\_no = new.course\_program\_no;

update member

set member\_course\_no = \_member\_course\_no

where member\_id = new.course\_member\_id;

end //

delimiter ;

show triggers;