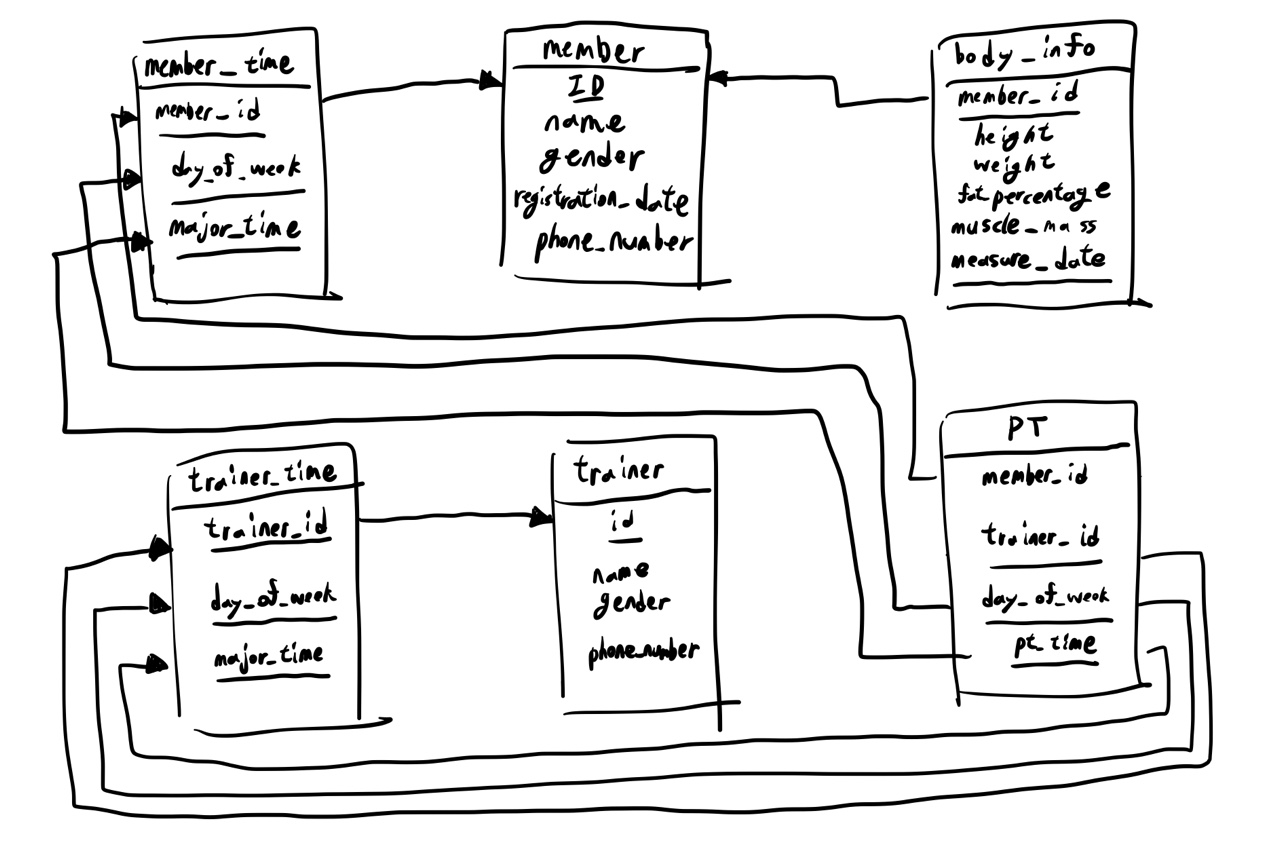
DB term project 보고서

2016160325 김준태

\*note: 프로그램 실행 시 코드상에서 psycopg2.connect()정보 local에 맞게 바꿔야합니다.

만든 application은 헬스장에서 회원들을 관리하는 application으로 회원정보 삽입, 수정, 삭제등을 할 수 있고 pt관계 확인이나 신체정보와 신체 변화 등을 확인할 수 있고 회원과 트레이너의 주요 시간대 정보를 이용해 pt추천을 하는 기능들이 있다.

- schema diagram



테이블은 위와 같이 6개이고 primary key, foreign key constraint는 위 diagram에 나타난 바와 같다. 각 table을 설명하도록 하겠다.

- member

텍스트이(가) 표시된 사진

자동 생성된 설명

회원 정보를 나타내 주는 table로 id가 primary key로 있고 그 외에 이름, 성별, 등록한 날짜, 전화번호 정보로 구성되어 있다.

- trainer

텍스트이(가) 표시된 사진

자동 생성된 설명

Trainer에 대한 정보로 id가 primary key이고 이름, 성별, 전화번호 정보로 구성되어 있다.

- body\_info

텍스트이(가) 표시된 사진

자동 생성된 설명

회원들의 신체정보에 관한 table로 (member\_id, measure\_date)세트가 primary key로 주어져 있다. 즉 한 회원이 하루에 한번만 신체정보를 등록할 수 있다는 가정이 있다. 회원 id, 키, 몸무게, 체지방량, 근육략, 측정날짜로 구성되어 있고 이때 member\_id는 foreign key로 member table의 id를 참조하고 있다. 또한 delete cascade를 걸어 놔서 특정 회원이 삭제되는 경우 그 회원의 body\_info도 같이 지워지도록 설정했다.

- member\_time

텍스트이(가) 표시된 사진

자동 생성된 설명

각 회원들이 헬스장을 이용할 수 있는 시간대 정보가 있다. 각 회원이 주로 이용할 수 있는 요일과 시간정보를 제공한다면 이러한 데이터가 있을 수 있다. 회원 id와 요일 주요시간대의 정보로 구성되어 있고 각 회원이 여러 요일의 여러 시간대에 헬스장이 이용 가능한 경우 여러 요일, 시간대조합을 등록할 수 있기 때문에 (회원 id, 요일, 시간대)set가 primary key로 주어지고 이때 회원 id는 member table의 id를 참조한다. 이때 delete cascade를 걸어서 특정 회원이 지워지면 해당 회원의 주요시간정보들인 member\_time정보도 같이 지워지도록 했다. 또한 헬스장이 07시부터 24시에 문을 닫는다고 가정하고 주요시간대는 6보다 크고 24보다 작아야 한다는 check조건을 걸어 놨다.

- trainer\_time

텍스트이(가) 표시된 사진

자동 생성된 설명

Member\_time과 비슷하다. 각 trainer가 헬스장을 이용하는 주요 시간대 정보가 주어져 있고 trainer id, 요일, 시간대정보로 구성되어 있다. 마찬가지로 주요시간대에 check조건이 걸려있고 (trainer id, 요일, 주요시간대)set가 primary key이고 trainer\_id가 foreign key로 trainer 테이블의 id를 참조하고 있다. Delete cascade를 걸어놨기 때문에 특정 trainer정보가 지워지면 해당 trainer의 trainer time의 관련정보들도 지워진다.

- PT

텍스트이(가) 표시된 사진

자동 생성된 설명

회원id, trainer id, pt요일, pt시간정보가 들어가 있다. 이때 특정 trainer는 특정 요일의 시간에는 특정 회원과만 pt를 할 수 있으므로 primary key를 (trainer\_id, 요일, 시간)으로 설정 했다.(반대로 (member\_id, 요일, 시간)을 primary key로 설정해도 괜찮다.) 이때 (member\_id, 요일, 시간)set가 member\_time table의 (member\_id, 요일, 시간)을 참조하도록 했고 delete cascade를 걸어서 member\_time의 관련 정보가 지워지면 같이 지워지도록 했다. 또한 (trainer\_id, 요일, 시간)set가 trainer\_time table의 (trainer\_id, 요일, 시간)을 참조하도록 했고 마찬가지로 delete cascade를 걸어서 trainer\_time table의 정보가 지워지면 해당 date를 참조하고 있던 data들도 pt table에서 같이 지워지도록 설정했다. 이러한 참조 관계를 갖는다는 의미는 각 trainer와 회원은 서로 각각 trainer\_time과 member\_time에 있는 시간대에만 pt를 한다는 의미이다.

- DB application기능 설명 및 관련 SQL문 설명

텍스트이(가) 표시된 사진

자동 생성된 설명

처음 들어가면 나타나는 화면이다. Member(register/delete/modify)를 누르면 다음과 같은 화면이 나온다.

테이블이(가) 표시된 사진

자동 생성된 설명

이때 먼저 보여지는 표는 현재 등록된 회원들의 정보이다. 사용된 SQL문은 “Select \* From member;”로 member table의 모든 정보를 가져오도록 했다.

표 위에 있는 입력 칸에 member\_id중 하나인 ‘M0000’을 치고 major\_time\_search버튼을 누르면 다음과 같은 표가 나온다.

테이블이(가) 표시된 사진

자동 생성된 설명

이때 이 표는 해당 id의 member의 헬스장 주요 시간대를 보여준다. 이때 사용된 SQL문은 “"Select member\_id, name, day\_of\_week, major\_time from member, member\_time where member\_id=id and member\_id=’m\_id’;"이고 의미는 member테이블과 member\_time테이블의 카르테지안 곱으로부터 각 두 table의 회원id가 일치하고 입력받은 m\_id에 해당하는 회원 id에 대해서만 회원 id, 회원 이름, 요일, 시간대 정보를 나타내도록 하는 쿼리문이다.

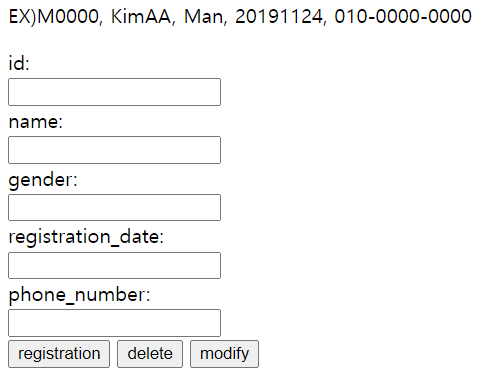
아래의 입력에 trainer id중 하나인 ‘T0000’을 치면 다음과 같은 결과가 나온다.

테이블이(가) 표시된 사진

자동 생성된 설명

마찬가지로 해당 trainer id를 갖는 trainer의 주요 헬스장 이용 정보로써 trainer id, 이름, 요일, 시간대 정보가 나온다. 이때 사용된 SQL문은 "Select trainer\_id, name, day\_of\_week, major\_time from trainer, trainer\_time where trainer\_id=id and trainer\_id=t\_id;"이고 의미는 trainer table, trainer\_time table로의 카르테지안 곱에서 각 table의 trainer id가 서로 같고 또 trainer id가 앞서 입력받은 id와 일치하는 경우 trainer\_id, 이름, 요일, 시간대 정보를 나타내도록 하라는 뜻이다.

다시 처음 member(register/delete/modify)버튼을 눌렀을 때 화면으로 돌아가면 표 아래 다음과 같은 입력버튼이 있는 것을 볼 수 있다.



각 버튼의 의미는 회원 등록, 삭제, 수정이다. 회원 정보를 다음과 같이 입력 후 registration해보면 표에 다음과 같이 M0020 회원이 추가가 된 것을 볼 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명 테이블이(가) 표시된 사진

자동 생성된 설명

이때 사용된 SQL문은 “insert into member values(mem\_id, name, gender, regist\_date, phone);"이고 이때 values()괄호 안에 있는 변수들은 모두 앞서 입력 받은 data들이 된다.

이후 다음과 같이 정보를 입력 후 modify버튼을 누르게 되면 표에서 해당 수정결과가 반영된 것을 볼 수 있다.(핸드폰 번호가 010-1111-1111에서 010-2222-2222로 변경된 것을 볼 수 있음)

테이블이(가) 표시된 사진

자동 생성된 설명

이때 사용된 SQL문은 "Update member set (id, name, gender, registration\_date, phone\_number) = (mem\_id', name, gender, regist\_date, phone) where id=mem\_id;"이다. 의미는 member table에서 입력받은 id(위 쿼리문에서 mem\_id)와 같은 id에대해서 (id, name, gender, registration\_date, phone\_number)정보를 (mem\_id', name, gender, regist\_date, phone)로 변경하라는 의미이다. 이때 (mem\_id', name, gender, regist\_date, phone)는 모두 입력받은 정보들이다.

이후 delete를 할 때는 회원 id만 입력해도 된다. 다음과 같이 입력 후 delete를 하면 M0020이 지워진 것을 확인할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명 테이블이(가) 표시된 사진

자동 생성된 설명

이때 사용된 SQL문은 “Delete From member where id=mem\_id;”이도 의미는 mem\_id와 같은 id를 가진 정보를 member table에서 지우라는 뜻이다. 여기서 mem\_id는 입력 받은 정보이다. 이때 알아 둘 점은 특정 회원을 지우면 지워진 특정 회원과 관련된 member\_time table, body\_info table, pt table의 정보들이 delete cascade로 인해 같이 지워지게 된다는 점이다.

다시 처음 화면에서 trainer\_info버튼을 누르면 다음과 같은 창이 나온다.

테이블이(가) 표시된 사진

자동 생성된 설명

해당 표는 현재 trainer들의 정보를 나타낸다. 이때 사용한 SQL문은 “Select \* From trainer;”로 trainer table로부터 모든 정보를 보여주게 된다. 위의 trainer major time search는 앞서 보여줬던 trainer의 id를 입력하면 해당 trainer의 주요시간대 정보를 보여주는 것과 동일한 기능이므로 다시 설명하지 않겠다.

다시 처음화면으로 돌아가서 pt\_info버튼을 누르면 다음과 같은 창이 나온다.

테이블이(가) 표시된 사진

자동 생성된 설명

우선 아래 표는 현재 등록되어 있는 pt정보들을 보여준다. 이때 사용한 SQL문은

"SELECT member.id as m\_id, member.name as m\_name, member.phone\_number as m\_phone,

trainer.id as t\_id, trainer.name as t\_name, trainer.phone\_number as t\_phone, PT.day\_of\_week, PT.pt\_time

From member, trainer, PT

Where member.id=pt.member\_id and trainer.id=pt.trainer\_id;"

이다. 이 SQL문은 member table, trainer table, PT table의 카르테지안 곱에서 member table의 id와 pt 테이블의 member\_id가 같고 trainer 테이블의 id와 pt table의 id가 같은 경우 member table의 id, 이름, 전화번호, trainer table의 id, 이름, 전화번호, pt table의 요일, 시간을 나타내도록 한다.

먼저 current pt info 아래 member\_id 아래있는 빈칸에 회원 id를 넣고 입력후 search 버튼을 누르면 해당회원의 pt정보가 아래와 같이 나타나게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명테이블이(가) 표시된 사진

자동 생성된 설명

이때 사용된 SQL문은 전체 pt정보를 보여주던 앞의 SQL문에 입력된 회원 id와 일치하는 경우에만 보여주도록 조건 하나만 추가한 것이고 다음과 같다.

"SELECT member.id as m\_id, member.name as m\_name, member.phone\_number as m\_phone,

trainer.id as t\_id, trainer.name as t\_name, trainer.phone\_number as t\_phone, PT.day\_of\_week, PT.pt\_time

From member, trainer, PT

Where member.id=pt.member\_id and trainer.id=pt.trainer\_id and member.id=m\_id;”

바로 앞서 전체 pt정보를 보여주는 쿼리의 where 절에 ‘member.id=m\_id’ 조건만 추가된 것이고 즉 앞의 창에서 보여준 pt정보에서 입력 받은 회원 id와 일치하는 경우만 따로 나타내 주라는 의미가 된다. 즉 member table, trainer table, PT table의 카르테지안 곱에서 member table의 id와 pt 테이블의 member\_id가 같고 trainer 테이블의 id와 pt table의 id가 같고 입력받은 회원 id와 member 테이블의 id가 같은 경우 member table의 id, 이름, 전화번호, trainer table의 id, 이름, 전화번호, pt table의 요일, 시간을 나타내도록 한다.

다시 돌아와서 위에 보면 pt\_recommendation 입력창이 있는데 이곳에 회원 id를 입력하고 search버튼을 누르게 되면 해당 회원의 주요 헬스장 이용 시간과 trainer의 헬스장 나오는 시간과 같고 또 그때 trainer가 현재 pt가 없다면 해당시간을 pt 추천시간으로써 추천해 주게 된다. 그러한 추천 시간정보를 나타내 주는 기능으로 다음과 같이 회원 id를 입력하면 추천정보가 나타난다.

텍스트이(가) 표시된 사진

자동 생성된 설명 테이블이(가) 표시된 사진

자동 생성된 설명

이때 사용한 SQL문은 우선 recommendation\_pt라는 이름의 view를 만들어주게 된다.

"create view recommendation\_pt as

select member\_id, trainer\_id, day\_of\_week, major\_time

from member\_time natural join trainer\_time

where (trainer\_id, day\_of\_week, major\_time) not in

(select trainer\_id, day\_of\_week, pt\_time from pt) and

member\_time.member\_id = m\_id;"

member\_time과 trainer\_time을 natural join해주게 되면 두 table에서 day\_of\_week과 major\_time이 attribute이름이 같아서 이 두개의 attribute가 겹치는 데이터들만 나타내 주게 된다. 즉 위에서 만든 recommendation\_pt view는 member\_time과 trainer\_time의 day\_of\_week과 major\_time이 같은 data들에서 (trainer\_id, day\_of\_week, major\_time)가 현재 pt table의 (trainer\_id, day\_of\_week, pt\_time)에 해당되지 않고 member\_time table의 회원id가 입력받은 회원 id와 같은 data들의 member\_id, trainer\_id, 요일, 시간순으로 나타내라는 뜻이다.

이러한 view를 만든 후 다음과 같은 SQL문을 실행했다.

"select member\_id, member.name, trainer\_id, trainer.name, day\_of\_week, major\_time,

member.phone\_number, trainer.phone\_number

from recommendation\_pt, member, trainer

where recommendation\_pt.member\_id=member.id and recommendation\_pt.trainer\_id=trainer.id;"

이 SQL문은 recommendation\_pt view의 member\_id와 member table의 id가 같고 recommendation\_pt view의 trainer\_id와 trainer table의 id가 같은 것들에 대해서 member table로부터 id, 이름, 전화번호를 가져오고 trainer table에서 id, 이름, 전화번호를 가져오고 recommendation\_pt view에서 요일과 시간대를 가져와서 나타내도록 하는 명령문이다. 이렇게 원하는 정보를 얻은 뒤 “drop view recommendation\_pt;”를 하도록 했는데 그 이유는 해당기능을 여러 번 이용할 경우 view를 매번 삭제 해주지 않으면 기능을 이용할 때마다 view를 생성하게 되는데 해당 이름의 view가 이미 있다는 error가 나기 때문에 매 기능을 이용할 때마다 view를 지우도록 했다.

이렇게 함으로써 현재 pt시간이안잡혀 있는 trainer들의 헬스장 출근 시간과 회원들이 주로 이용하는 시간대를 매칭해서 pt 추천 시간대를 알려줄 수 있다.

다시 처음 화면으로 돌아와서 body\_info버튼을 누르면 다음과 같은 창이 나타나게 된다.

테이블이(가) 표시된 사진

자동 생성된 설명

표는 각 회원들의 id, 키, 몸무게, 체지방량, 골격근량, 신체측정 날짜를 나타낸다. 이때 사용된 SQL문은 "SELECT \* From body\_info;"로 body\_info table의 모든 정보를 나타내도록 하는 것이다.

위의 member\_id칸에 회원 id를 넣고 search버튼을 누르게 되면 다음과 같은 정보가 나타난다.

테이블이(가) 표시된 사진

자동 생성된 설명

테이블이(가) 표시된 사진

자동 생성된 설명

이 기능은 해당 회원이 처음 신체를 측정한 날짜와 가장 최근에 측정한 날의 신체정보를 보여주고 그 차이를 보여주게 된다. 사용한 SQL문들은 다음과 같고 우선 f\_l\_body라는 이름의 view를 생성해주도록 한다.(f: first, l: last의 약자)

"create view f\_l\_body as

Select member\_id as m\_id, min(measure\_date) as f\_date, max(measure\_date) as l\_date

From body\_info group by member\_id;"

이 view는 body\_info table로부터 member\_id로 그룹 정렬을 한 뒤 신체측정 날짜의 최소와 최대값을 나타내도록 하는 view이다. 이때 최소값인 날짜가 가장 처음 측정한 날짜가 되고 최대값인 날짜가 가장 최근에 측정한 날짜가 된다.

이후 f\_body라는 이름의 view를 만들어주는데 이 view는 입력 받은 회원의 처음 날짜의 신체정보를 나타내 준다.

"create view f\_body as

Select f\_date, member\_id, height, weight, fat\_percentage, muscle\_mass

from f\_l\_body, body\_info

where f\_l\_body.f\_date=body\_info.measure\_date and f\_l\_body.m\_id=body\_info.member\_id and

f\_l\_body.m\_id=m\_id;"

이 f\_body라는 view는 f\_l\_body view와 body\_info table로부터 f\_l\_body의 f\_date와 body\_info의 measure\_date가 일치하고 f\_l\_body의 회원 id와 body\_info의 회원 id가 일치하고 f\_l\_body의 회원 id와 입력받은 회원id정보가 일치하는 경우에 f\_date(처음측정날짜), 회원 id, 키, 몸무게, 체지방률, 골격근량을 나타내도록하는 query이다.

이후 “select \* from f\_body;”를 통해 f\_body의 모든 정보를 나타내도록 한 것이 바로 first date body info의 표가 된다.

이후 이것과 비슷하게 또 l\_body라는 view를 만들어주게 된다.

"create view l\_body as

Select l\_date, member\_id, height, weight, fat\_percentage, muscle\_mass

from f\_l\_body, body\_info

where f\_l\_body.l\_date=body\_info.measure\_date and f\_l\_body.m\_id=body\_info.member\_id and

f\_l\_body.m\_id=m\_id;"

앞서 f\_body와 비슷하다. L\_body view는 f\_l\_body view와 body\_info table로부터 f\_l\_body의 l\_date와 body\_info의 measure\_date가 일치하고 f\_l\_body의 회원 id와 body\_info의 회원 id가 일치하고 f\_l\_body의 회원 id와 입력받은 회원id정보가 일치하는 경우에 l\_date(가장최근측정날짜), 회원 id, 키, 몸무게, 체지방률, 골격근량을 나타내도록하는 query이다.

이후 “select \* from l\_body;”를 통해 l\_body의 모든 정보를 나타내도록 한 것이 바로 last date body info의 표가 된다.

마지막 변화량 표의 경우 처음 측정한 신체와 가장 최근 측정한 신체가 얼마나 달라졌는지를 보여주는데 양수면 그만큼 증가했다는 것이고 음수면 그만큼 감소했다는 뜻이다. 해당 표를 보이기 위해 사용된 SQL문은 다음과 같다.

Select f\_body.member\_id, (l\_body.height-f\_body.height), (l\_body.weight-f\_body.weight),

(l\_body.fat\_percentage-f\_body.fat\_percentage), (l\_body.muscle\_mass-f\_body.muscle\_mass)

from f\_body, l\_body;"

의미는 f\_body view와 l\_body view로부터 f\_body의 회원 id와 l\_body의 (키, 몸무게, 체지방률, 골격근량)값에서 f\_body의 (키, 몸무게, 체지방률, 골격근량)값을 각각 뺀 정보들을 나타내 주라는 뜻이다.

이렇게 해당기능들을 이용한 후에 앞서 pt추천의 경우와 마찬가지 이유로 생성한 모든 view들은 drop해주도록 했다.